Referências Bibliográficas

[1] TEMPORAL, A.. **Compressão com perdas, de imagens obtidas por satelites de sensoriamento remoto, para transmissão em canal com ruido.** Dissertação de Mestrado, Pontificia Universidade Católica do Rio de Janeiro, Set.2002.

[2] ELIAS, P.. **Coding for two noisy channels**. Information Theory, Third London Symposium, p. 61–76, 1955.

[3] SANCHEZ PAIBA, FRANKLIN. **Códigos fontanais para canais com apagamento**. Dissertação de Mestrado, Pontificia Universidade Católica do Rio de Janeiro, Jul.2008.

[4] HANG, W.. **Hardware design for [lt] coding.** Dissertação de Mestrado, Delft University of Technology, 2006.

[5] LUBY, M.. **LT codes**. Proc. of the 43rd Annual IEEE Symp. on Foundation of Comp. Sc., p. 271–280, Novembro 2002.

[6] MACKAY, D. J. C.. **Information Theory, Inference, and Learning Algorithms**. Cambridge University Press, 2003.

[7] MAYMOUNKOV, P.; MAZIERES, D.. **Rateless codes and big downloads**. In: Proc. of the 2nd International Workshop Peer-to-Peer System, 2003.

[8] PALANKI, R.; YEDIDIA, S.. **Rateless codes on noisy channels**. Mitsubshi Electric Research Laboratories, Abril 2004.

[9] REED, I. S.; SOLOMON, G.. **Polynomial Codes Over Certain Finite Fields**. J. Soc. Indust. Appl. Math, Vol. 8:pag. 300–304, 1960.

[10] SAID, A.; A., P. W.. **A new, fast, and efficient image codec based on set partitioning in hierarchichal trees**. IEEE Trans. Circuits Syst. Video Technol, Vol.10:pag. 926–943, Junho 1996.

[11] SASAKI, C.; HASEGAWA, T. ; KOBAYASHI, S.. **On Unicast based Recovery for Multicast Content Distribution considering XOR-FEC**. Asia-Pacific Conference on Communications, Outubro 2005.

[12] SHAPIRO, J. M.. **Embedded image coding zerotrees of wavelet coefficients**. IEEE Trans. Signal Proceessing, Vol.41:pag. 3445–3462, Decembro 1993.

[13] SHERWOOD, P. G.; ZEGER, K.. **Progressive image coding for noisy channels**. IEEE Signal Processing Letters, vol. 4:pag. 189–191, Julho 1997.

[14] TEE, R.; NGUYEN, T.; YANG, L. ; HANZO, L.. **Serially Concatenated Luby Transform Coding And Bit-Interleaved Coded Modulation Using Iterative Decoding For The Wireless Internet**. Proceedings of VTC 2006 Spring, Melbourne, vol. 138:pag. 177–182, Maio 2006.

[15] LUBY, M.; MITZENMACHER, M. ; SHOKROLLAHI, A.. **Efficient Erasure Correction Codes**. IEEE Trans. on Information Theory, Vol. 47:pag. 569–584, Fevereiro 2001.

[16] WICKER, S. B.. **Error Control Systems for Digital Communication and Storage**. Prentice Hall, Upper Saddle River, NJ07458 - USA, 1995.

A
Codificação em bloco

A.1
Código em bloco lineares

Considerando um código em bloco **C** consistente em n *-tuples* $(c_0, c_1, c_2, ..., c_{n-1})$ de símbolos pertencentes ao GF(q). C é um código q-ário linear se e somente se C forma um subespaço vetorial sobre GF(q).

A dimensão de um código linear é a dimensão correspondente ao espaço vetorial.

Uma notação conveniente é usualmente usada para referenciar códigos lineares : Um código de comprimento n e dimensão k é chamado um código (n, k) . Um código (n, k) com símbolos em GF(q) possui um total de q^k palavras código de comprimento n. Para os códigos lineares se tem um conjunto de propriedades que são propriedades dos espaços vetoriais.

- A combinação linear de quaisquer conjunto de palavras código é também uma palavra código
- A distancia minima de um código linear é igual ao peso mais baixo da palavra código distinta da palavra código composta por zeros.
- Os padrões de erros não detetáveis são independentes da palavra código transmitida e sempre consistem em um conjunto de de palavra código diferentes da palavra código composta por zeros.

Seja $\mathbf{g_0}, \mathbf{g_1}, \mathbf{g_2}, ..., \mathbf{g_{k-1}}$ uma base para as palavras código de um código C q-ário (n, k) se tem então uma representação $\mathbf{c} = \mathbf{a_0 g_0} + \mathbf{a_1 g_1} + ... + \mathbf{a_{k-1} g_{k-1}}$ para cada palavra código c $\in$ **C**. Assim cada combinação linear dos elementos da base resulta uma palavra código, isto é um mapeamento entre o conjunto de k blocos de símbolos $(a_0, a_1, a_2, ..., a_{k-1})$ sobre GF(q) e as palavras código em **C**. Uma Matriz **G** é construída tomando os vetores da base como linhas como se mostra na equação (A-1).

$$G = \begin{pmatrix} \mathbf{g_0} \\ \mathbf{g_1} \\ \vdots \\ \mathbf{g_{k-1}} \end{pmatrix} = \begin{pmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,n-1} \\ g_{1,0} & g_{1,1} & \cdots & g_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{pmatrix} \tag{A-1}$$

A matriz **G** é a matriz geradora do código **C**. esta pode ser usada para codificar blocos de informação de k-símbolos da seguinte forma: seja **m**= $(m_0, m_1, m_2, ...m_{k-1})$ um bloco q-ário de dados sem codigo, a mensagem codificada c se obtém com se mostra na equacão (A-2)

$$mG = (m_0, m_1, m_2, m_3, ..., m_{k-1}) \begin{pmatrix} \mathbf{g_0} \\ \mathbf{g_1} \\ \vdots \\ \mathbf{g_{k-1}} \end{pmatrix} \tag{A-2}$$

$$= m_0\mathbf{g_0} + m_1\mathbf{g_1} + \ldots + m_{k-1}\mathbf{g_{k-1}} = \mathbf{c} \tag{A-3}$$

Um código **C** q-ário de comprimento n é um subespaço vetorial V contido no espaço de todas as n-tuples sobre GF(q). neste contexto pode se falar do espaço dual de **C** contido em V. O espaço dual de um código linear é conhecido como o código dual de **C** e é denotado por $\mathbf{C}^{\perp}$ este é um espaço vetorial de dimensão $(n-k)$. Assim que uma base $\mathbf{h_0}, \mathbf{h_1}, \mathbf{h_2}, ..., \mathbf{h_{n-k-1}}$ para $\mathbf{C}^{\perp}$ pode ser usada para construir uma matriz H para cheque de paridade (A-4).

$$H = \begin{pmatrix} \mathbf{h_0} \\ \mathbf{h_1} \\ \vdots \\ \mathbf{h_{n-k-1}} \end{pmatrix} = \begin{pmatrix} h_{0,0} & h_{0,1} & \cdots & h_{0,n-1} \\ h_{1,0} & h_{1,1} & \cdots & h_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n-k-1,0} & h_{n-k-1,1} & \cdots & h_{n-k-1,n-1} \end{pmatrix} \tag{A-4}$$

Pode ser provado que um vetor c é uma palavra código in **C** se e solo se $\mathbf{cH^T} = \mathbf{0}$, este fato pode ser usado para a identificação de uma palavra código errada numa transmissão onde possam acontecer erros, assim quando no receptor se encontra que o produto de uma palavra código recebida e matriz cheque de paridade transposta seja distinta de zero, quer dizer que aquela palavra código foi corrompida pelo ruido e por tanto ela se encontra errada.

A matriz geradora e a matriz cheque de paridade simplificam consideravelmente a codificação no transmissor e a detecção no receptor . ambas

operações são reduzidas a uma simples multiplicação de matrices , mais obviamente se precisa de tabelas de look up armazenadas em memoria.

O problema de recuperar os blocos de informação a partir de uma palavra código pode ser simplificado com o uso de uma codificação sistemática . Considerando uma matriz geradora G de um código linear **C**. Usando eliminação gaussiana e reorganizando as colunas é sempre possível obter uma matriz geradora da forma da equacão (A-5)

$$\mathbf{G} = [\mathbf{P}|\mathbf{I_k}] = \begin{pmatrix} p_{0,0} & p_{0,1} & \dots & p_{0,n-k-1} & | & 1 & 0 & 0 & \dots & 0 \\ p_{1,0} & p_{1,1} & \dots & p_{1,n-k-1} & | & 0 & 1 & 0 & \dots & 0 \\ p_{2,0} & p_{2,1} & \dots & p_{2,n-k-1} & | & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & | & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{k-1,0} & p_{k-1,1} & \dots & p_{k-1,n-k-1} & | & 0 & 0 & 0 & \dots & 1 \end{pmatrix} \quad \text{(A-5)}$$

Quando um bloco de informação é codificado usando uma matriz geradora sistemática a informação original queda contida em as ultimas k coordenadas de palavra código resultante.

$$\begin{aligned} \mathbf{c} &= m\mathbf{G} && \text{(A-6)} \\ &= (m_0, m_1, m_2, m_3, ..., m_{k-1})[\mathbf{P}|\mathbf{I_k}] \\ &= [c_0, c_1, c_2, c_3, ..., c_{n-k-1}|m_0, m_1, m_2, m_3, ..., m_{k-1}] && \text{(A-7)} \end{aligned}$$

Depois da decodificação , os últimos k símbolos são removidos da palavra código selecionada e passados onde vai ser processada a informação.

Dada uma matriz geradora da forma da equação (A-5), a correspondente matriz cheque de paridade pode ser obtida como se mostra na equação (A-8).

$$\begin{aligned} \mathbf{H} &= [I_{n-k}| - P^T] && \text{(A-8)} \\ &= \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & | & -p_{0,0} & -p_{1,0} & -p_{2,0} & \dots & -p_{k-1,0} \\ 0 & 1 & 0 & \dots & 0 & | & -p_{0,1} & -p_{1,1} & -p_{2,1} & \dots & -p_{k-1,1} \\ 0 & 0 & 1 & \dots & 0 & | & -p_{0,2} & -p_{1,2} & -p_{2,2} & \dots & -p_{k-1,2} \\ \vdots & \vdots & \vdots & \ddots & 1 & | & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 & | & -p_{0,n-k-1} & -p_{1,n-k-1} & -p_{2,n-k-1} & \dots & -p_{k-1,n-k-1} \end{pmatrix} \end{aligned}$$

A.2
Códigos Hamming

Desenvolvidos por Hamming quando trabalhava nos laboratórios da telefônica Bell, na década de 1940, são a classe mais importante de códigos lineares binários , sua primeira aplicação foi feita na telefonia de longa distancia
.

Os parâmetros de desempenho para a familia de códigos de hamming são usualmente expressados como função de um inteiro $m \geq 2$.

- Comprimento do código: $n = 2^m - 1$
- Numero de símbolos de informação: $k = 2^m - m - 1$,
- Numero de símbolos de paridade: $n - k = m$
- Capacidade de correção de erros: $t = 1$

A matriz cheque de paridade para um código binário Hamming resulta simples de construir.Para um código Hamming de comprimento $(2^m - 1)$, só se tem que construir uma matriz cujas colunas consistem em todas as m-tuplas binárias não zero. Para um código Hamming $(15, 11)$um exemplo da matriz cheque de paridade se mostra em na equação A-9 o ordem das colunas é arbitrário, a matriz geradora correspondente apresenta-se em na equação A-10.

Um exame na matriz cheque de paridade na figura A-9 mostra que o código Hamming (15,11) só pode corrigir um erro na palavra código. Em geral pode se notar que a suma das m-tuples mais pequenas distintas a palavra zero suma sempre três, por tanto este código tem uma minima distancia três e tem capacidade de correção de um erro.

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \tag{A-9}$$

$$\mathbf{G} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(A-10)}$$